

Approximated Structured Prediction for Learning Large Scale Graphical Models

Tamir Hazan
tamir@ttic.edu

TTI Chicago
6045 S. Kenwood Ave.,
Chicago, IL 60637 USA

Raquel Urtasun
rurtasun@ttic.edu

TTI Chicago
6045 S. Kenwood Ave.,
Chicago, IL 60637 USA

Abstract

In this paper we propose an approximated structured prediction framework for large scale graphical models and derive message-passing algorithms for learning their parameters efficiently. We first relate CRFs and structured SVMs and show that in CRFs a variant of the log-partition function, known as soft-max, smoothly approximates the hinge loss function of structured SVMs. We then propose an intuitive approximation for the structured prediction problem, using duality, based on local entropy approximations and derive an efficient message-passing algorithm that is guaranteed to converge to the optimum for concave entropy approximations. Unlike existing approaches, this allows us to learn efficiently graphical models with cycles and very large number of parameters. We demonstrate the effectiveness of our approach in an image denoising task. This task was previously solved by sharing parameters across cliques. In contrast, our algorithm is able to efficiently learn large number of parameters resulting in orders of magnitude better prediction.

1. Introduction

Unlike standard supervised learning problems which involve simple scalar outputs, structured prediction deals with structured outputs such as sequences, grids, or more general graphs. Ideally, one would want to make joint predictions on the structured labels instead of simply predicting each element independently, as this additionally accounts for the statistical correlations between label elements, as well as between training examples and their labels. These properties make structured prediction appealing for a wide range of applications such as image segmentation, image denoising, sequence labeling and natural language parsing.

Several structured prediction models have been recently proposed, including log-likelihood models such as conditional random fields (CRFs, Lafferty et al. (2001)), and structured support vector machines (structured SVMs) such as maximum-margin Markov networks (M3Ns Taskar et al. (2004)) and structured output learning (Tsochantaridis et al. (2006)). For CRFs, learning is done by minimizing a convex function composed of a negative log-likelihood loss and a regularization term. Learning structured SVMs is done by minimizing the convex regularized structured hinge loss.

Despite the convexity of the objective functions, finding the optimal parameters of these models can be computationally expensive since it involves exponentially many labels. When the label structure corresponds to a tree, learning can be done efficiently by using belief propagation as a subroutine; The sum-product algorithm is typically used in CRFs and the max-product algorithm in structured SVMs. When the label structure corresponds to a general graph, one cannot compute the objective nor the gradient exactly, and usually resorts to approximate inference algorithms, e.g. Finley and Joachims (2008); Taskar et al. (2006) for structured SVMs and Taskar et al. (2002); Levin and Weiss (2006); Yanover et al. (2007) for CRFs. However, the approximate inference algorithms are computationally too expensive to be used as a subroutine of the learning algorithm, therefore they cannot be applied efficiently for large scale structured prediction problems. Also, it is not clear how to define a stopping criteria as it approximates the objective and gradient, and as a consequence the objective does not monotonically decrease, and may result in poor approximations.

In this paper we propose an approximated structured prediction framework for large scale graphical models and derive message-passing algorithms for learning their parameters efficiently. We relate CRFs and structured SVMs, and show that in CRFs a variant of the log-partition function, known as soft-max, smoothly approximates the hinge loss function of structured SVMs. We then propose an intuitive approximation for the structured prediction problem, using duality, based on a local entropy approximation and derive an efficient message-passing algorithm that is guaranteed to converge to the optimum for concave entropy approximations. Unlike existing approaches, this allows us to learn efficiently graphical models with cycles and very large number of parameters. We demonstrate the effectiveness of our approach in an image denoising task. This task was previously solved by sharing parameters across cliques. In contrast, our algorithm is able to efficiently learn large number of parameters resulting in orders of magnitude better prediction.

The rest of the paper is organized as follows. In Section 2 we review regularized loss minimization focusing on its most common models, CRFs and structured SVMs. We relate CRFs and structured SVMs in Section 2.1, and present the corresponding graphical models in Section 2.2. We present our approximate prediction framework in Section 3, derive a message-passing algorithm to solve the approximated problem efficiently in Section 4, and show our experimental evaluation in Section 5.

2. Regularized Loss Minimization

Consider a supervised learning setting with objects $x \in X$ and labels $y \in \mathcal{Y}$. In structured prediction the labels may be sequences, trees, grids, or other high-dimensional objects with internal structure. Consider a function $\Phi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^d$ that maps (x, y) pairs to feature vectors. Our goal is to construct a linear prediction rule

$$y_{\theta}(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \theta^{\top} \Phi(x, y)$$

with parameters $\theta \in \mathbb{R}^d$, such that $y_{\theta}(x)$ is a good approximation to the true label of x . The parameters θ are typically learned by minimizing the regularized loss

$$\sum_{(x,y) \in \mathcal{S}} \ell(\theta, x, y) + \frac{C}{p} \|\theta\|_p^p, \quad (1)$$

defined over a training set \mathcal{S} . The function ℓ measures the loss incurred in using $y_{\theta}(x)$ to predict the label of x , given that the true label is y .

In this paper we focus on structured SVMs and CRFs which are the most common structured prediction models. The first definition of structured SVMs used the structured hinge loss, originally introduced by Taskar et al. (2004)

$$\ell_{\text{hinge}}(\theta, x, y) = \max_{\hat{y} \in \mathcal{Y}} \left\{ e(y, \hat{y}) + \theta^\top \Phi(x, \hat{y}) - \theta^\top \Phi(x, y) \right\}$$

where $e(y, \hat{y})$ is some non-negative measure of error when predicting \hat{y} instead of y as the label of x . We assume that $e(y, y) = 0$, so that no loss is incurred for correct prediction. This loss function corresponds to a maximum-margin approach that explicitly penalizes training examples (x, y) for which $\theta^\top \Phi(x, y) < e(y, y_{\theta}(x)) + \theta^\top \Phi(x, y_{\theta}(x))$.

The second loss function that we consider is based on log-linear models, and is commonly used in CRFs (Lafferty et al. (2001)). Let the conditional distribution be

$$p(\hat{y}|\theta_{x,y}) = \frac{1}{Z(x, y)} \exp \left(e_y(\hat{y}) + \theta^\top \Phi(x, \hat{y}) \right), \quad Z(x, y) = \sum_{\hat{y} \in \mathcal{Y}} \exp \left(e_y(\hat{y}) + \theta^\top \Phi(x, \hat{y}) \right)$$

where $e_y(\hat{y}) = e(y, \hat{y})$ corresponds to a prior distribution, and $Z(x, y)$ is the partition function. The loss function is then the negative log-likelihood under the parameters θ

$$\ell_{\log}(\theta, x, y) = \ln \frac{1}{p(y|\theta_{x,y})}.$$

In structured SVMs and CRFs a convex loss function and a convex regularization are minimized.

2.1 One parameter extension of CRFs and Structured SVMs

In CRFs one aims to minimize the regularized negative log-likelihood of the conditional distribution $p(\hat{y}|\theta_{x,y})$ which decomposes into the log-partition $Z(x, y)$ and the linear term $\theta^\top \Phi(x, y)$. Hence the problem of minimizing the regularized loss in (1) with the loss function ℓ_{\log} can be written as

$$(\text{CRF}) \quad \min_{\theta} \left\{ \sum_{(x,y) \in \mathcal{S}} \ln Z(x, y) - \mathbf{d}^\top \theta + \frac{C}{p} \|\theta\|_p^p \right\},$$

where $(x, y) \in \mathcal{S}$ ranges over training pairs and $\mathbf{d} = \sum_{(x,y) \in \mathcal{S}} \Phi(x, y)$ is the vector of empirical means. In gradient based methods, a coordinate θ_r is updated in the direction of the negative gradient, for some step size η . The gradient of the log-partition function corresponds to the probability distribution $p(\hat{y}|\theta_{x,y})$, and the direction of descent takes the form

$$\sum_{(x,y) \in \mathcal{S}} \sum_{\hat{y} \in \mathcal{Y}} p(\hat{y}|\theta_{x,y}) \phi_r(x, \hat{y}) - d_r + |\theta_r|^{p-1} \text{sign}(\theta_r).$$

Structured SVMs aim at minimizing the regularized hinge loss $\ell_{\text{hinge}}(\theta, x, y)$, which measures the loss of the label $y_{\theta}(x)$ that most violates the training pair $(x, y) \in \mathcal{S}$ by more

than $e(y, y_{\theta}(x))$. Since $y_{\theta}(x)$ is independent of the training label y , the structured SVM program takes the form:

$$(\text{structured SVM}) \quad \min_{\theta} \left\{ \sum_{(x,y) \in \mathcal{S}} \max_{\hat{y} \in \mathcal{Y}} \left\{ e(y, \hat{y}) + \theta^{\top} \Phi(x, \hat{y}) \right\} - \mathbf{d}^{\top} \theta + \frac{C}{p} \|\theta\|_p^p \right\},$$

where $(x, y) \in \mathcal{S}$ ranges over the training pairs, and \mathbf{d} is the vector of empirical means. The structured SVM objective involves the max function, hence it is not smooth. However every convex function $f(\theta)$ has a *subdifferential*, denoted by $\partial f(\theta)$, which is the family of *subgradients*, i.e. supporting hyperplanes to its epigraph at $(\theta, f(\theta))$. The subdifferential generalizes the concept of the gradient since a convex function is smooth if and only if its subdifferential consists of a single vector, i.e. its gradient (cf. Bertsekas et al. (2003), Theorem 4.2.2). Danskin's theorem (cf. Bertsekas et al. (2003), Theorem 4.5.1) states that the subdifferential of the max function corresponds to the probability distributions $p(y^* | \theta_{x,y})$ over the optimal set $\mathcal{Y}^* = \operatorname{argmax}_{\hat{y} \in \mathcal{Y}} \{e(y, \hat{y}) + \theta^{\top} \Phi(x, \hat{y})\}$, therefore the subdifferential of the structured SVM takes the form

$$\sum_{(x,y) \in \mathcal{S}} \sum_{y^* \in \mathcal{Y}^*} p(y^* | \theta_{x,y}) \phi_r(x, y^*) - d_r + |\theta_r|^{p-1} \cdot \operatorname{sign}(\theta_r)$$

Unlike the smooth case, a negative subgradient not necessarily points towards a direction of descent, therefore subgradient methods are usually not monotonically decreasing, and depend on the step size. Their optimal solution is taken from the algorithm sequence.

In the following we deal with both structured prediction tasks (i.e., structured SVMs and CRFs) as two instances of the same framework, by extending the partition function to norms, namely $Z_{\epsilon}(x, y) = \|\exp(e_y(\hat{y}) + \theta^{\top} \Phi(x, \hat{y}))\|_{1/\epsilon}$, where the norm is computed for the vector ranging over $\hat{y} \in \mathcal{Y}$. Using the norm formulation we move from the partition function, for $\epsilon = 1$, to the maximum over the exponential function for $\epsilon = 0$. Equivalently, we relate the log-partition and the max-function by the soft-max function

$$\ln Z_{\epsilon}(x, y) = \epsilon \ln \sum_{\hat{y} \in \mathcal{Y}} \exp \left(\frac{e_y(\hat{y}) + \theta^{\top} \Phi(x, \hat{y})}{\epsilon} \right) \quad (2)$$

For $\epsilon = 1$ the soft-max function reduces to the log-partition function, and for $\epsilon = 0$ it reduces to the max-function. Moreover, for $\epsilon \rightarrow 0$ the soft-max function is a smooth approximation of the max-function, in the same way the $\ell_{1/\epsilon}$ -norm is a smooth approximation of the ℓ_{∞} -norm. This smooth approximation of the max-function is used in different areas of research, e.g. Vontobel and Koetter (2006); Johnson et al. (2007). We thus define the structured prediction problem as

$$(\text{structured-prediction}) \quad \min_{\theta} \left\{ \sum_{(x,y) \in \mathcal{S}} \ln Z_{\epsilon}(x, y) - \mathbf{d}^{\top} \theta + \frac{C}{p} \|\theta\|_p^p \right\}, \quad (3)$$

which is a one-parameter extension of CRFs and structured SVMs, i.e., $\epsilon = 1$ and $\epsilon = 0$ respectively. Similarly to CRFs and structured SVMs (Lebanon and Lafferty (2002); Ratliff

et al. (2006)), one can use gradient methods to optimize structured prediction. The gradient of θ_r takes the form

$$\sum_{(x,y) \in \mathcal{S}} \sum_{\hat{y} \in \mathcal{Y}} p_\epsilon(\hat{y}|\theta_{x,y}) \phi_r(x, \hat{y}) - d_r + |\theta_r|^{p-1} \text{sign}(\theta_r), \quad (4)$$

where

$$p_\epsilon(\hat{y}|\theta_{x,y}) = \frac{1}{Z_\epsilon(x, y)^{1/\epsilon}} \exp \left(\frac{e_y(\hat{y}) + \theta^\top \Phi(x, \hat{y})}{\epsilon} \right) \quad (5)$$

is a probability distribution over the possible labels $\hat{y} \in \mathcal{Y}$. When $\epsilon \rightarrow 0$ this probability distribution gets concentrated around its maximal values, since all its elements are raised to the power of a very large number (i.e. $1/\epsilon$) normalized by $Z_\epsilon(x, y)$. Therefore for $\epsilon = 0$ we get a structured SVM subgradient.

One can compute the soft-max (2) and the probability $p_\epsilon(\hat{y}|\theta_{x,y})$ in (5) using variational methods (cf. Wainwright and Jordan (2008), Theorem 8.1) since

$$\ln Z_\epsilon(x, y) = \max_{p(\hat{y}) \in \Delta_{\mathcal{Y}}} \sum_{\hat{y} \in \mathcal{Y}} p(\hat{y}) \left(e_y(\hat{y}) + \theta^\top \Phi(x, \hat{y}) \right) + \epsilon H(\mathbf{p}), \quad (6)$$

where $\Delta_{\mathcal{Y}}$ is the set of all probability distribution over \mathcal{Y} , namely $p(\hat{y}) \geq 0$, $\sum_{\hat{y}} p(\hat{y}) = 1$, and $H(\mathbf{p})$ is the entropy of the probability distribution $p(\hat{y})$. One can verify that the distribution $p_\epsilon(\hat{y}|\theta_{x,y})$ in (5) is the optimal argument of the program in (6), by differentiating and finding the vanishing point of the gradient.

2.2 Structured Prediction in Graphical Models

In many real-life applications the labels $y \in \mathcal{Y}$ are n -tuples, $y = (y_1, \dots, y_n)$ for $y_v \in \mathcal{Y}_v$, hence there are exponentially many labels in \mathcal{Y} . The feature maps usually describe relations between subsets of label elements $y_\alpha \subset \{y_1, \dots, y_n\}$, and local evidence on single label elements y_v , namely

$$\phi_r(x, \hat{y}_1, \dots, \hat{y}_n) = \sum_{v \in V_{r,x}} \phi_{r,v}(x, \hat{y}_v) + \sum_{\alpha \in E_{r,x}} \phi_{r,\alpha}(x, \hat{y}_\alpha). \quad (7)$$

Each feature $\phi_r(x, \hat{y})$ can be described by its *factor graph* $G_{r,x}$, a bipartite graph with one set of nodes corresponding to $V_{r,x}$ and the other set corresponds to $E_{r,x}$. An edge connects a single label node $v \in V_{r,x}$ with a subset of label nodes $\alpha \in E_{r,x}$ if and only if $y_v \in y_\alpha$. In the following we consider the factor graph $G = \cup G_{r,x}$ which is the union of all factor graphs. We denote by $N(v)$ and $N(\alpha)$ the set of neighbors of v and α respectively, in the factor graph G . For clarity in the presentation we consider fully factorized priors

$$e_y(\hat{y}_1, \dots, \hat{y}_n) = \sum_{v=1}^n e_{y,v}(\hat{y}_v),$$

although our derivation naturally extends to any graphical model representing the interactions $e_y(\hat{y})$.

The local structure on the features (7) in turn induces a local structure on the learning problem (3). In particular, the gradient (4) takes the local structure

$$\sum_{(x,y) \in \mathcal{S}} \left(\sum_{\hat{y}_v \in \mathcal{Y}_v} p_\epsilon(\hat{y}_v | \boldsymbol{\theta}_{x,y}) \phi_{r,v}(x, \hat{y}_v) + \sum_{\hat{y}_\alpha \in \mathcal{Y}_\alpha} p_\epsilon(\hat{y}_\alpha | \boldsymbol{\theta}_{x,y}) \phi_{r,\alpha}(x, \hat{y}_\alpha) \right) - d_r + |\theta_r|^{p-1} \text{sign}(\theta_r),$$

which involves the averages of the local features $\phi_{r,v}(x, \hat{y}_v), \phi_{r,\alpha}(x, \hat{y}_\alpha)$ with respect to marginal probabilities $p_\epsilon(\hat{y}_v | \boldsymbol{\theta}_{x,y})$ and $p_\epsilon(\hat{y}_\alpha | \boldsymbol{\theta}_{x,y})$. Moreover, using the variational method one can observe a local structure on the linear terms of the soft-max function

$$\ln Z_\epsilon(x, y) = \max_{p(\hat{y}) \in \Delta_{\mathcal{Y}}} \sum_{\hat{y}_v \in \mathcal{Y}_v} p(\hat{y}_v) \phi_v(x, \hat{y}_v) + \sum_{\hat{y}_\alpha \in \mathcal{Y}_\alpha} p(\hat{y}_\alpha) \phi_\alpha(x, \hat{y}_\alpha) + \epsilon H(\mathbf{p}),$$

where $p(\hat{y}_v), p(\hat{y}_\alpha)$ are the marginal probabilities of $p(\hat{y})$. Also, we used a short notation for the $\boldsymbol{\theta}$ weighted features in the graphical model $\phi_v(x, \hat{y}_v) = e_y(\hat{y}_v) + \sum_{r:v \in V_{r,x}} \theta_r \phi_{r,v}(x, \hat{y}_v)$ and $\phi_\alpha(x, \hat{y}_\alpha) = \sum_{r:\alpha \in E_{r,x}} \theta_r \phi_{r,\alpha}(x, \hat{y}_\alpha)$. We note that in general the soft-max function does not admit a fully local structure due to the entropy function.

When the factor graph has no cycles, the probability distribution can be represented by a multiplication of its marginal probabilities $p(\hat{y}) = \prod_\alpha p(\hat{y}_\alpha) \prod_v p(\hat{y}_v)^{1-|N(v)|}$. Therefore the entropy $H(\mathbf{p})$ can be represented by sum of local entropies over the marginal probabilities, $\sum_\alpha H(p(\hat{y}_\alpha)) - \sum_v (1 - |N(v)|) H(p(\hat{y}_v))$, which is known as the Bethe entropy. In this case, using the variational method (6), the soft-max function $\ln Z_\epsilon(x, y)$ admits the fully local structure

$$\max_{p(\hat{y}) \in \Delta_{\mathcal{Y}}} \sum_{\hat{y}_v \in \mathcal{Y}_v} p(\hat{y}_v) \phi_v(x, \hat{y}_v) + \sum_{\hat{y}_\alpha \in \mathcal{Y}_\alpha} p(\hat{y}_\alpha) \phi_\alpha(x, \hat{y}_\alpha) + \epsilon \left(\sum_\alpha H(p(\hat{y}_\alpha)) - \sum_v (1 - |N(v)|) H(p(\hat{y}_v)) \right)$$

and the marginal probabilities $p_\epsilon(\hat{y}_v | \boldsymbol{\theta}_{x,y}), p_\epsilon(\hat{y}_\alpha | \boldsymbol{\theta}_{x,y})$ are the optimal arguments of the local variational program (Yedidia et al. (2005)). This describes a way to compute the soft-max function in the structured prediction objective (3) and the marginal probabilities for the gradient (4) without considering exponentially many elements $\hat{y} \in \mathcal{Y}$ for graphs without cycles. This enables to explicitly determine a step size to ensure the change of θ_r in the negative gradient direction reduces the structured prediction objective.

In general, when the graphical model has cycles, the structured prediction objective is exponentially hard to compute since the soft-max, $\ln Z_\epsilon(x, y)$, considers exponentially many elements $\hat{y} \in \mathcal{Y}$. Similarly, the gradient involves summing over exponentially many elements since it requires the marginal probabilities $p_\epsilon(\hat{y}_v | \boldsymbol{\theta}_{x,y})$ and $p_\epsilon(\hat{y}_\alpha | \boldsymbol{\theta}_{x,y})$. In the following we describe the approximate inference framework which is typically used to approximate the soft-max and the marginal probabilities. The approximate inference framework is based on the variational method in (6) which derives the soft-max by the marginal probabilities when the factor graph has no cycles. The main idea is to replace the marginal probabilities $p(\hat{y}_v), p(\hat{y}_\alpha)$ with beliefs $b_v(\hat{y}_v), b_\alpha(\hat{y}_\alpha)$ and the entropy term by sum of local entropies. The

approximation of (6) takes the form:

$$\begin{aligned} \ln Z_\epsilon(x, y) \approx & \max_{v, \hat{y}_v} \sum b_v(\hat{y}_v) \phi_v(x, \hat{y}_v) + \sum_{\alpha, \hat{y}_\alpha} b_\alpha(\hat{y}_\alpha) \phi_\alpha(x, \hat{y}_\alpha) + \epsilon \left(\sum_\alpha c_\alpha H(b_\alpha) + \sum_v c_v H(b_v) \right), \\ \text{subject to } & b_v(\hat{y}_v) \in \Delta_{\mathcal{Y}_v}, \quad b_\alpha(\hat{y}_\alpha) \in \Delta_{\mathcal{Y}_\alpha}, \quad \sum_{\hat{y}_\alpha \setminus \hat{y}_v} b_\alpha(\hat{y}_\alpha) = b_v(\hat{y}_v), \end{aligned} \quad (8)$$

where $\Delta_{\mathcal{Y}_v}$ is the set of probability distributions over \mathcal{Y}_v , i.e. $b_v(\hat{y}_v) \geq 0$, $\sum_{\hat{y}_v} b_v(\hat{y}_v) = 1$, and $\Delta_{\mathcal{Y}_\alpha}$ is the set of probability distributions over \mathcal{Y}_α .

The local entropy approximation $\sum_\alpha c_\alpha H(b_\alpha) + \sum_v c_v H(b_v)$ in (8) is known as the fractional entropy approximation of Wiergerinck and Heskes (2003). When the graphical model has no cycles, and the fractional entropy weights are chosen according to the Bethe entropy, i.e. $c_\alpha = 1, c_v = 1 - |N(v)|$, then variational program in (8) is equivalent to the one in (6); it gives an exact characterization of the soft-max, and its optimal beliefs are the true marginal probabilities. However, when the graphical model has cycles this variational program is an approximation of the soft-max and the marginal probabilities, and no guarantees on the quality of the approximation are known so far.

To compute the soft-max and the marginal probabilities, $p_\epsilon(\hat{y}_v | \theta_{x,y})$ and $p_\epsilon(\hat{y}_\alpha | \theta_{x,y})$, exponentially many labels have to be considered. This is in general computationally prohibitive, but when the factor graph has no cycles this can be done efficiently by the belief propagation algorithms which send messages along the edges of the factor graph (Pearl (1988)). However, in the presence of cycles inference can only be approximated (refeq:approx). In the following we present a message-passing algorithm for solving the approximate inference problem, (cf. Wainwright et al. (2005a,b); Heskes (2006); Meltzer et al. (2009); Hazan and Shashua (2009)). To follow the product form of beliefs propagation algorithms we set $\bar{\phi}_v(\hat{y}_v) = \exp(\phi_v(\hat{y}_v))$, and $\bar{\phi}_\alpha(\hat{y}_\alpha) = \exp(\phi_\alpha(\hat{y}_\alpha))$. In this case the messages can be computed as

$$m_{\alpha \rightarrow v}(\hat{y}_v) = \left\| \bar{\phi}_\alpha(\hat{y}_\alpha) \prod_{u \in N(\alpha) \setminus v} n_{u \rightarrow \alpha}(\hat{y}_u) \right\|_{1/\epsilon c_\alpha}, \quad n_{v \rightarrow \alpha}(\hat{y}_v) \propto \frac{\left(\bar{\phi}_v(\hat{y}_v) \prod_{\beta \in N(v) \setminus \alpha} m_{\beta \rightarrow v}(\hat{y}_v) \right)^{c_\alpha / \hat{c}_v}}{m_{\alpha \rightarrow v}(\hat{y}_v)},$$

where the norm is computed for the vector ranging over \hat{y}_α while holding \hat{y}_v fixed, and $\hat{c}_v = c_v + \sum_{\alpha \in N(v)} c_\alpha$ and \propto indicates that the vector can be normalized. After convergence, one can infer the beliefs by

$$b_v(\hat{y}_v) \propto \left(\bar{\phi}_v(\hat{y}_v) \prod_{\alpha \in N(v)} m_{\alpha \rightarrow v}(\hat{y}_v) \right)^{1/\epsilon c_v}, \quad b_\alpha(\hat{y}_\alpha) \propto \left(\bar{\phi}_\alpha(\hat{y}_\alpha) \prod_{u \in N(\alpha)} n_{u \rightarrow \alpha}(\hat{y}_u) \right)^{1/\epsilon c_\alpha},$$

When the factor graph has no cycles, one can use the above message-passing algorithm with the Bethe entropy, i.e. $c_\alpha = 1$ and $c_v = 1 - |N(v)|$ to compute in linear time the soft-max, $\ln Z_\epsilon(x, y)$, and the marginal probabilities, $p_\epsilon(\hat{y}_v | \theta_{x,y})$ and $p_\epsilon(\hat{y}_\alpha | \theta_{x,y})$, therefore it can be used as a subroutine to compute the structure prediction objective (3) and gradient (4).

Focusing on the Bethe entropy, this message-passing algorithm is the ϵ parameter extension of the belief propagation algorithms, which includes as special cases the sum-product for $\epsilon = 1$ and the max-product for $\epsilon = 0$. It can be shown that for every positive ϵ this belief propagation extension solves exactly (6) for graphs without cycles, since by using a change of variables it reduces to the sum-product algorithm for $\epsilon > 0$.

When the factor graph has cycles this message-passing algorithm has in general no guarantees for convergence (unless $c_\alpha, c_v > 0$), nor on the number of iterations, nor on the quality of the solution, and it is used as an *approximation* for the soft-max and the marginal probabilities. Therefore, there are two main problems when dealing with graphs with cycles and approximate inference: efficiency and accuracy. For graphs with cycles there are no guarantees on the number of steps the message-passing algorithm requires till convergence, therefore it is computationally costly to run it as a subroutine. Also, as these message-passing algorithms have no guarantees on the quality of their solution, the gradient and the objective function can only be approximated. Therefore one cannot verify if the update of θ_r in the negative approximated gradient direction decreased or increased the structured prediction objective. In general, this heuristic results in an algorithm without a clear stopping criteria.

In contrast, in this work we propose to approximate the structured prediction problem and to efficiently solve the approximated problem exactly using message-passing. This allows us to efficiently learn graphical models with large number of parameters.

3. Approximate Structured Prediction

The structured prediction objective in (3) and its gradients defined in (4) cannot be computed efficiently for general graphs since both involve computing the soft-max function and the marginal probabilities, which take into account exponentially many elements $\hat{y} \in Y$. In the following we suggest an intuitive approximation for structured prediction, based on its dual formulation.

We believe that a main difficulty in dealing with convex programs, is that special care has to be taken to consider the set of feasible solutions, when constructing the dual function. We find it simpler to describe the primal program using extended real-valued convex functions, which are functions that can get the value of infinity. Intuitively, by using these functions we can ignore their domains, simplifying the derivations. The dual programs of extended real valued convex functions $g : \mathbb{R}^k \rightarrow \mathbb{R}$ are formulated in terms of their conjugate dual

$$g^*(\mathbf{z}) = \max_{\boldsymbol{\mu}} \left\{ \boldsymbol{\mu}^\top \mathbf{z} - g(\boldsymbol{\mu}) \right\}.$$

Throughout this work we use the following duality theorem, known as *Fenchel duality*, Fenchel (1951); Rockafellar (1970); Bertsekas et al. (2003):

Theorem 1 *Let Φ be a $k_1 \times k_2$ matrix, and let $\mathbf{p}, \mathbf{e} \in \mathbb{R}^{k_2}$ and $\boldsymbol{\theta}, \mathbf{d} \in \mathbb{R}^{k_1}$ be vectors. The following are primal-dual programs:*

$$\begin{aligned} (\text{Primal}) \quad & \min_{\boldsymbol{\theta}} \left\{ f(\Phi^\top \boldsymbol{\theta} + \mathbf{e}) - \mathbf{d}^\top \boldsymbol{\theta} + h(-\boldsymbol{\theta}) \right\} \\ (\text{Dual}) \quad & \max_{\mathbf{p}} \left\{ -f^*(\mathbf{p}) + \mathbf{p}^\top \mathbf{e} - h^*(\Phi \mathbf{p} - \mathbf{d}) \right\} \end{aligned}$$

Proof: We use Lagrange duality theorem, minimizing the function $f(\boldsymbol{\mu} + \mathbf{e}) - \mathbf{d}^\top \boldsymbol{\theta} + h(-\boldsymbol{\theta})$ subject to the constraints $\boldsymbol{\mu} = \Phi^\top \boldsymbol{\theta}$. These equality constraints hold for every coordinate indexed by $\{1, \dots, k_2\}$, therefore correspond to k_2 Lagrange multipliers $\mathbf{p} \in \mathbb{R}^{k_2}$. The Lagrangian takes the form

$$L(\boldsymbol{\mu}, \boldsymbol{\theta}, \mathbf{p}) = f(\boldsymbol{\mu} + \mathbf{e}) - \mathbf{d}^\top \boldsymbol{\theta} + h(-\boldsymbol{\theta}) - \mathbf{p}^\top (\boldsymbol{\mu} - \Phi^\top \boldsymbol{\theta}).$$

By minimizing with respect to the primal variables $\min_{\boldsymbol{\mu}, \boldsymbol{\theta}} L(\boldsymbol{\mu}, \boldsymbol{\theta}, \mathbf{p})$ we get the dual function above. \square

Since the conjugate dual of the soft-max is the entropy barrier, it follows that the dual program for structured prediction is governed by the entropy function of the probabilities $p_{x,y}(\hat{y})$. The following duality formulation is known for CRFs when $\epsilon = 1$ with ℓ_2^2 regularization, and for structured SVM when $\epsilon = 0$ with ℓ_2^2 regularization (Lebanon and Lafferty (2002); Taskar et al. (2004); Collins et al. (2008)). Here we derive the dual program for every ϵ and every ℓ_p^p regularization using conjugate duality:

Claim 1 *The dual program of the structured prediction program in (3) takes the form*

$$\max_{p_{x,y}(\hat{y}) \in \Delta_{\mathcal{Y}}} \sum_{(x,y) \in \mathcal{S}} \left(\epsilon H(\mathbf{p}_{x,y}) + \mathbf{p}_{x,y}^\top \mathbf{e}_y \right) - \frac{C^{1-q}}{q} \left\| \sum_{(x,y) \in \mathcal{S}} \sum_{\hat{y} \in \mathcal{Y}} p_{x,y}(\hat{y}) \Phi(x, \hat{y}) - \mathbf{d} \right\|_q^q,$$

where $\Delta_{\mathcal{Y}}$ is the probability simplex over \mathcal{Y} , $H(\mathbf{p}_{x,y}) = -\sum_{\hat{y}} p_{x,y}(\hat{y}) \ln p_{x,y}(\hat{y})$ is the entropy function and $\mathbf{p}_{x,y}^\top \mathbf{e}_y = \sum_{\hat{y}} p_{x,y}(\hat{y}) e_y(\hat{y})$.

Proof: The proof follows the one of Theorem 1. We first describe an equivalent program to the one in (3) by adding variables $\mu(x, \hat{y})$ instead of $\boldsymbol{\theta}^\top \Phi(x, \hat{y})$ to decouple the soft-max from the regularization.

$$\min_{\substack{\boldsymbol{\theta}, \mu(x, \hat{y}) \\ \mu(x, \hat{y}) = \boldsymbol{\theta}^\top \Phi(x, \hat{y})}} \left\{ \sum_{(x,y) \in \mathcal{S}} \epsilon \ln \sum_{\hat{y}} \exp \frac{e_y(\hat{y}) + \mu(x, \hat{y})}{\epsilon} - \mathbf{d}^\top \boldsymbol{\theta} + \frac{C}{p} \|\boldsymbol{\theta}\|_p^p \right\},$$

To maintain consistency, we add the constraints $\mu(x, \hat{y}) = \boldsymbol{\theta}^\top \Phi(x, \hat{y})$, for every $(x, y) \in \mathcal{S}$ and every $\hat{y} \in \mathcal{Y}$. We compute the Lagrangian by adding the Lagrange multipliers $p_{x,y}(\hat{y})$

$$L() = \sum_{(x,y) \in \mathcal{S}} \epsilon \ln \sum_{\hat{y} \in \mathcal{Y}} \exp \frac{e_y(\hat{y}) + \mu(x, \hat{y})}{\epsilon} - \mathbf{d}^\top \boldsymbol{\theta} + \frac{C}{p} \|\boldsymbol{\theta}\|_p^p - \sum_{(x,y) \in \mathcal{S}, \hat{y} \in \mathcal{Y}} p_{x,y}(\hat{y}) \left(\mu(x, \hat{y}) - \boldsymbol{\theta}^\top \Phi(x, \hat{y}) \right).$$

The dual function is a function of the Lagrange multipliers, and it is derived by minimizing the Lagrangian, namely $q(\mathbf{p}_{x,y}) = \min_{\boldsymbol{\mu}, \boldsymbol{\theta}} L(\boldsymbol{\mu}, \boldsymbol{\theta}, \mathbf{p}_{x,y})$. The dual function can be written as

$$\sum_{(x,y)} \min_{\mu(x, \hat{y})} \left\{ \epsilon \ln \sum_{\hat{y}} \exp \frac{e_y(\hat{y}) + \mu(x, \hat{y})}{\epsilon} - \sum_{\hat{y}} \mu(x, \hat{y}) p_{x,y}(\hat{y}) \right\} + \min_{\boldsymbol{\theta}} \left\{ \frac{C}{p} \|\boldsymbol{\theta}\|_p^p - \boldsymbol{\theta}^\top \left(\sum_{(x,y), \hat{y}} p_{x,y}(\hat{y}) \Phi(x, \hat{y}) - \mathbf{d} \right) \right\}$$

hence it is composed from the conjugate dual of the soft-max and the conjugate dual of the ℓ_p^p norm. Recall that the conjugate dual for the soft-max is the entropy barrier $\epsilon H(p_{x,y})$ over the set of probability distributions $\Delta_{\mathcal{Y}}$ (cf. Wainwright and Jordan (2008) Theorem 8.1). Also, the linear shift $e_y(\hat{y})$ of the soft-max argument results in the linear shift of the conjugate dual, thus we get the first part of the dual function $\sum(\epsilon H(\mathbf{p}_{x,y}) + \mathbf{e}_y^\top \mathbf{p}_{x,y})$. Similarly, the conjugate dual of $\frac{1}{p} \|\boldsymbol{\theta}\|_p^p$ is $\frac{1}{q} \|\mathbf{z}\|_q^q$ for the dual norm $1/p + 1/q = 1$ (cf. Rockafellar (1970), page 106), where in our case $\mathbf{z} = \sum_{(x,y),\hat{y}} p_{x,y}(\hat{y}) \Phi(x, \hat{y}) - \mathbf{d}$. \square

When $\epsilon = 1$ the CRF dual program reduces to the well-known duality relation between the log-likelihood and the entropy. When $\epsilon = 0$ we obtain the dual formulation of structured SVM which emphasizes the duality relation between the max-function and the probability simplex. In general, Claim 1 describes the relation between the soft-max function and the entropy barrier over the probability simplex.

The dual formulation in Claim 1 gives more information on the structured prediction program in (3), in particular, it demonstrates different connections between structured SVMs and CRFs. Both models try to fit a probability distribution $\mathbf{p}_{x,y}$ to a prior \mathbf{e}_y , while matching the empirical means to be as close as possible to the learned model means, $\mathbf{d} \approx \sum_{(x,y) \in \mathcal{S}} \sum_{\hat{y} \in \mathcal{Y}} p_{x,y}(\hat{y}) \Phi(x, \hat{y})$. However, in CRFs the $\mathbf{p}_{x,y}$ are chosen with respect to a KL-divergence from the prior $\exp(-\mathbf{e}_y)$, whereas in structured SVMs they are chosen with respect to the inner product $\mathbf{p}_{x,y}^\top \mathbf{e}_y$.

Intuitively, this one-parameter extension implies that we can approximate structured SVMs by solving CRFs with prior $\exp(-\mathbf{e}_y/\epsilon)$, and weighting the regularization by $C^{1-q}/(\epsilon q)$, while taking $\epsilon \rightarrow 0$. This is equivalent to minimizing the dual program

$$\epsilon \cdot \max_{\mathbf{p}_{x,y}} \sum_{(x,y) \in \mathcal{S}} \left(H(\mathbf{p}_{x,y}) + \mathbf{p}_{x,y}^\top \frac{\mathbf{e}_y}{\epsilon} \right) - \frac{C^{1-q}}{\epsilon q} \left\| \sum_{(x,y) \in \mathcal{S}} \sum_{\hat{y} \in \mathcal{Y}} p_{x,y}(\hat{y}) \Phi(x, \hat{y}) - \mathbf{d} \right\|_q^q$$

For example, considering the zero-one loss, the prior suggests the dual optimal solution is a distribution which is concentrated around the training label, while weighting the regularization differently. Although this approach is algorithmically unstable for $\epsilon \rightarrow 0$ compared to the formulation in (3), it may give useful intuition on how to relate both approaches by considering different weights on the priors and regularizations.

The dual program in Claim 1 considers the probabilities $p_{x,y}(\hat{y})$ over exponentially many labels $\hat{y} \in \mathcal{Y}$, as well as their entropies $H(\mathbf{p}_{x,y})$. However, when we take into account the graphical model $G_{r,x}$ imposed by the features we observe that the linear terms in the dual formulation consider the marginal probabilities $p_{x,y}(\hat{y}_v)$ and $p_{x,y}(\hat{y}_\alpha)$. We thus propose to replace the marginal probabilities with their corresponding beliefs $b_{x,y,v}(\hat{y}_v)$, $b_{x,y,\alpha}(\hat{y}_\alpha)$, and to replace the entropy term by the sum of local entropies $\sum_\alpha c_\alpha H(\mathbf{b}_{x,y,\alpha}) + \sum_v c_v H(\mathbf{b}_{x,y,v})$.

This results in the following approximation of the structured prediction problem

(approximated structured prediction - dual)

$$\begin{aligned}
 & \max_{b_{x,y,v}(\hat{y}_v), b_{x,y,\alpha}(\hat{y}_\alpha)} \sum_{(x,y) \in \mathcal{S}} \left(\sum_{\alpha \in E} \epsilon c_\alpha H(\mathbf{b}_{x,y,\alpha}) + \sum_{v \in V} \epsilon c_v H(\mathbf{b}_{x,y,v}) + \sum_{v \in V, \hat{y}_v} b_{x,y,v}(\hat{y}_v) e_{y,v}(\hat{y}_v) \right) \\
 & - \frac{C^{1-q}}{q} \sum_r \left| \sum_{\substack{(x,y) \in \mathcal{S}, \\ v \in V_{r,x}, \hat{y}_v}} b_{x,y,v}(\hat{y}_v) \phi_{r,v}(x, \hat{y}_v) + \sum_{\substack{(x,y) \in \mathcal{S}, \\ \alpha \in E_{r,x}, \hat{y}_\alpha}} b_{x,y,\alpha}(\hat{y}_\alpha) \phi_{r,\alpha}(x, \hat{y}_\alpha) - d_r \right|^q \\
 & \text{subject to} \\
 & b_{x,y,v}(\hat{y}_v) \in \Delta_{\mathcal{Y}_v}, \quad b_{x,y,\alpha}(\hat{y}_\alpha) \in \Delta_{\mathcal{Y}_\alpha}, \quad \sum_{\hat{y}_\alpha \setminus \hat{y}_v} b_{x,y,\alpha}(\hat{y}_\alpha) = b_{x,y,v}(\hat{y}_v) \quad (9)
 \end{aligned}$$

Whenever $\epsilon, c_v, c_\alpha \geq 0$, the approximated dual (9) is concave and its dual is a convex primal program. By deriving the dual of (9) we obtain our approximated structured prediction, for which we construct an efficient algorithm in Section 4.

Theorem 2 *The approximation of the structured prediction program in (3) takes the form*

$$\begin{aligned}
 & \min_{\lambda_{x,y,v \rightarrow \alpha}, \boldsymbol{\theta}} \sum_{(x,y) \in \mathcal{S}, v} \epsilon c_v \ln \sum_{\hat{y}_v} \exp \left(\frac{e_y(\hat{y}_v) + \sum_{r: v \in V_{r,x}} \theta_r \phi_{r,v}(x, \hat{y}_v) - \sum_{\alpha \in N(v)} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{\epsilon c_v} \right) \\
 & + \sum_{(x,y) \in \mathcal{S}, \alpha} \epsilon c_\alpha \ln \sum_{\hat{y}_\alpha} \exp \left(\frac{\sum_{r: \alpha \in E_r} \theta_r \phi_{r,\alpha}(x, \hat{y}_\alpha) + \sum_{v \in N(\alpha)} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{\epsilon c_\alpha} \right) - \mathbf{d}^\top \boldsymbol{\theta} - \frac{C}{p} \|\boldsymbol{\theta}\|_p^p
 \end{aligned}$$

Proof: The proof follows the one of Theorem 1. We first describe an equivalent program to the one in (9) by adding variables z_r to decouple the entropies from the moment matching constraints.

$$\max \sum_{(x,y) \in \mathcal{S}} \left(\sum_{\alpha \in E} \epsilon c_\alpha H(\mathbf{b}_{x,y,\alpha}) + \sum_{v \in V} \epsilon c_v H(\mathbf{b}_{x,y,v}) + \sum_{v \in V, \hat{y}_v} b_{x,y,v}(\hat{y}_v) e_{y,v}(\hat{y}_v) \right) - \frac{C^{1-q}}{q} \|\mathbf{z} - \mathbf{d}\|_q^q$$

subject to the beliefs marginalization constraints, and the consistency constraints

$$z_r = \sum_{(x,y) \in \mathcal{S}, v \in V_{r,x}, \hat{y}_v} b_{x,y,v}(\hat{y}_v) \phi_{r,v}(x, \hat{y}_v) + \sum_{(x,y) \in \mathcal{S}, \alpha \in E_{r,x}, \hat{y}_\alpha} b_{x,y,\alpha}(\hat{y}_\alpha) \phi_{r,\alpha}(x, \hat{y}_\alpha).$$

We derive the Lagrangian by introducing the Lagrange multipliers $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ for every marginalization constraint $\sum_{\hat{y}_\alpha \setminus \hat{y}_v} b_{x,y,\alpha}(\hat{y}_\alpha) = b_{x,y,v}(\hat{y}_v)$, and Lagrange multipliers θ_r for

every equality constraint involving z_r . In particular, the Lagrangian has the form:

$$\begin{aligned}
L() = & \sum_{(x,y) \in \mathcal{S}} \left(\sum_{\alpha \in E} \epsilon c_\alpha H(\mathbf{b}_{x,y,\alpha}) + \sum_{v \in V} \epsilon c_v H(\mathbf{b}_{x,y,v}) + \sum_{v \in V, \hat{y}_v} b_{x,y,v}(\hat{y}_v) e_{y,v}(\hat{y}_v) \right) - \frac{C^{1-q}}{q} \|\mathbf{z} - \mathbf{d}\|_q^q \\
& + \sum_r \theta_r \left(\sum_{(x,y) \in \mathcal{S}, v \in V_r, \hat{y}_v} b_{x,y,v}(\hat{y}_v) \phi_{r,v}(x, \hat{y}_v) + \sum_{(x,y) \in \mathcal{S}, \alpha \in E_r, \hat{y}_\alpha} b_{x,y,\alpha}(\hat{y}_\alpha) \phi_{r,\alpha}(x, \hat{y}_\alpha) - z_r \right) \\
& + \sum_{v, \alpha \in N(v), \hat{y}_v} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v) \left(\sum_{\hat{y}_\alpha \setminus \hat{y}_v} b_{x,y,\alpha}(\hat{y}_\alpha) - b_{x,y,v}(\hat{y}_v) \right)
\end{aligned}$$

We obtain the dual function by minimizing the beliefs over their compact domain, i.e.

$$q(\boldsymbol{\lambda}_{x,y,v \rightarrow \alpha}, \boldsymbol{\theta}) = \max_{b_{x,y,v}(\hat{y}_v) \in \Delta_{\mathcal{Y}_v}, b_{x,y,\alpha}(\hat{y}_\alpha) \in \Delta_{\mathcal{Y}_\alpha}} L(\mathbf{b}_{x,y,v}, \mathbf{b}_{x,y,\alpha}, \boldsymbol{\lambda}_{x,y,v \rightarrow \alpha}, \boldsymbol{\theta}),$$

Deriving the dual by minimizing over the compact set of beliefs enables us to obtain an *unconstrained* dual, which corresponds to the approximated structured prediction program. The dual function is described by the conjugate dual functions:

$$\begin{aligned}
& \sum_{(x,y) \in \mathcal{S}, v} \max_{\mathbf{b}_{x,y,v} \in \Delta_{\mathcal{Y}_v}} \left\{ \epsilon c_v H(\mathbf{b}_{x,y,v}) + \sum_{\hat{y}_v} b_{x,y,v}(\hat{y}_v) \left(e_y(\hat{y}_v) + \sum_{r: v \in V_r} \theta_r \phi_{r,v}(x, \hat{y}_v) - \sum_{\alpha \in N(v)} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v) \right) \right\} \\
& + \sum_{(x,y) \in \mathcal{S}, \alpha} \max_{\mathbf{b}_{x,y,\alpha} \in \Delta_{\mathcal{Y}_\alpha}} \left\{ \epsilon c_\alpha H(\mathbf{b}_{x,y,\alpha}) + \sum_{\hat{y}_\alpha} b_{x,y,\alpha}(\hat{y}_\alpha) \left(\sum_{r: \alpha \in E_r} \theta_r \phi_{r,\alpha}(x, \hat{y}_\alpha) + \sum_{v \in N(\alpha)} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v) \right) \right\} \\
& + \max_{\mathbf{z}} \left\{ -\frac{C^{1-q}}{q} \|\mathbf{z} - \mathbf{d}\|_q^q - \mathbf{z}^\top \boldsymbol{\theta} \right\}
\end{aligned}$$

Its final form is derived similarly to Claim 1, where we show that the conjugate dual of the entropy barrier is the soft-max function and the conjugate dual of the ℓ_q^q is the ℓ_p^p . \square

Comparing the structured prediction in (3) to the approximated structured prediction in Theorem 2, we conclude that introducing beliefs to approximated the dual (9) is equivalent to decomposing the soft-max over $\hat{y}_1, \dots, \hat{y}_n$ (which is exponential in n) into the sum of soft-max over \hat{y}_v and \hat{y}_α . This approximation introduces the messages $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ that are the Lagrange multipliers which enforce the local marginalization constraints over the beliefs.

For the particular case of CRFs (i.e., $\epsilon = 1$) the approximated structured prediction decomposes the log-partition function into a sum of efficiently computable log-partition functions, while maintaining consistencies using the messages $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$. Similarly, for $\epsilon = 0$, the approximated structured prediction induces an approximation for structured SVMs, decomposing the max-function into a sum of local max-functions. The consistency between the separate max function is maintained by the messages $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$. For $\epsilon \rightarrow 0$ the approximated structured prediction introduces a smooth approximation for the approximated structured SVMs. This is useful from an algorithmic point of view where one can use gradient methods which are in general faster than subgradient methods.

4. Message-Passing Algorithm for Approximated Structured Prediction

In the following we describe a block coordinate descent algorithm for the approximated structured prediction program of Theorem 2. Coordinate descent methods are appealing as they optimize a small number of variables while holding the rest fixed, therefore they can be performed efficiently and can be easily parallelized. Since the primal program is lower bounded by the dual program, the primal objective function is guaranteed to converge.

We begin by describing how to find the optimal set of variables related to a node v in the graphical model, namely $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ for every $\alpha \in N(v)$, every \hat{y}_v and every $(x, y) \in \mathcal{S}$.

Lemma 3 *Given a vertex v in the graphical model, the optimal $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ for every $\alpha \in N(v)$, $\hat{y}_v \in \mathcal{Y}_v$, $(x, y) \in \mathcal{S}$ in the approximated program of Theorem 2 satisfies*

$$\begin{aligned} \mu_{x,y,\alpha \rightarrow v}(\hat{y}_v) &= \epsilon c_\alpha \ln \left(\sum_{\hat{y}_\alpha \setminus \hat{y}_v} \exp \left(\frac{\sum_{r:\alpha \in E_{r,x}} \theta_r \phi_{r,\alpha}(x, \hat{y}_\alpha) + \sum_{u \in N(\alpha) \setminus v} \lambda_{x,y,u \rightarrow \alpha}(\hat{y}_u)}{\epsilon c_\alpha} \right) \right) \\ \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v) &= \frac{c_\alpha}{\hat{c}_v} \left(e_{y,v}(\hat{y}_v) + \sum_{r:v \in V_{r,x}} \theta_r \phi_{r,v}(x, \hat{y}_v) + \sum_{\beta \in N(v)} \mu_{x,y,\beta \rightarrow v}(\hat{y}_v) \right) - \mu_{x,y,\alpha \rightarrow v}(\hat{y}_v) + c_{x,y,v \rightarrow \alpha} \end{aligned}$$

for every constant $c_{x,y,v \rightarrow \alpha}^1$, where $\hat{c}_v = c_v + \sum_{\alpha \in N(v)} c_\alpha$. In particular, if either ϵ and/or c_α are zero then $\mu_{x,y,\alpha \rightarrow v}$ corresponds to the ℓ_∞ norm and can be computed by the max-function. Moreover, if either ϵ and/or c_α are zero in the objective, then the optimal $\lambda_{x,y,v \rightarrow \alpha}$ can be computed for any arbitrary $c_\alpha > 0$, and similarly for $c_v > 0$.

Proof: For a given x, y and v , optimizing $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ for every $\alpha \in N(v)$ and $\hat{y}_v \in \mathcal{Y}_v$ while holding the rest of the variables fixed, reduces the problem to

$$\begin{aligned} \min_{\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)} \quad & \epsilon c_v \ln \sum_{\hat{y}_v} \exp \left(\frac{e_y(\hat{y}_v) + \sum_{r:v \in V_{r,x}} \theta_r \phi_{r,v}(x, \hat{y}_v) - \sum_{\alpha \in N(v)} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{\epsilon c_v} \right) \\ & + \sum_{\alpha \in N(v)} \epsilon c_\alpha \ln \sum_{\hat{y}_\alpha} \exp \left(\frac{\sum_{r:\alpha \in E_r} \theta_r \phi_{r,\alpha}(x, \hat{y}_\alpha) + \sum_{v \in N(\alpha)} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{\epsilon c_\alpha} \right) \end{aligned}$$

Let

$$\mu_{x,y,\alpha \rightarrow v}(\hat{y}_v) = c_\alpha \ln \sum_{\hat{y}_\alpha \setminus \hat{y}_v} \exp \left(\frac{\sum_{r:\alpha \in E_r} \theta_r \phi_{r,\alpha}(x, \hat{y}_\alpha) + \sum_{u \in N(\alpha) \setminus v} \lambda_{x,y,u \rightarrow \alpha}(\hat{y}_u)}{\epsilon c_\alpha} \right),$$

and also $\phi_{x,y,v}(\hat{y}_v) = e_y(\hat{y}_v) + \sum_{r:v \in V_{r,x}} \theta_r \phi_{r,v}(x, \hat{y}_v)$. We find the optimal $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ whenever the gradient vanishes, i.e.

$$0 = \nabla \left\{ \epsilon c_\alpha \ln \sum_{\hat{y}_v} \exp \left(\frac{\mu_{x,y,\alpha \rightarrow v}(\hat{y}_v) + \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{\epsilon c_\alpha} \right) + \epsilon c_v \ln \sum_{\hat{y}_v} \exp \left(\frac{\phi_{x,y,v}(\hat{y}_v) - \sum_{\alpha \in N(v)} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{\epsilon c_v} \right) \right\}$$

1. For numerical stability in our algorithm we set $c_{x,y,v \rightarrow \alpha}$ such that $\sum_{\hat{y}_v} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v) = 0$

Taking the vanishing point of the gradient we derive two probabilities over \hat{y}_v that need to be the same, namely

$$\frac{\exp\left(\frac{\mu_{x,y,\alpha \rightarrow v}(\hat{y}_v) + \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{\epsilon c_\alpha}\right)}{\sum_{\tilde{y}_v} \exp\left(\frac{\mu_{x,y,\alpha \rightarrow v}(\tilde{y}_v) + \lambda_{x,y,v \rightarrow \alpha}(\tilde{y}_v)}{\epsilon c_\alpha}\right)} = \frac{\exp\left(\frac{\phi_{x,y,v}(\hat{y}_v) - \sum_{\beta \in N(v)} \lambda_{x,y,v \rightarrow \beta}(\hat{y}_v)}{\epsilon c_v}\right)}{\sum_{\tilde{y}_v} \exp\left(\frac{\phi_{x,y,v}(\tilde{y}_v) - \sum_{\beta \in N(v)} \lambda_{x,y,v \rightarrow \beta}(\tilde{y}_v)}{\epsilon c_v}\right)}.$$

For simplicity we need to consider only the numerator, while taking one degree of freedom in the normalization. Taking log of the numerator we deduce that the gradient vanishes if the following holds

$$\tilde{c}_{x,y,v \rightarrow \alpha} + \frac{\mu_{x,y,\alpha \rightarrow v}(\hat{y}_v) + \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{c_\alpha} = \frac{\phi_{x,y,v}(\hat{y}_v) - \sum_{\beta \in N(v)} \lambda_{x,y,v \rightarrow \beta}(\hat{y}_v)}{c_v}. \quad (10)$$

Multiplying both sides of the equation by $c_v c_\alpha$, and summing both sides with respect to $\beta \in N(v)$ gives

$$\tilde{c}_{x,y,v \rightarrow \alpha} + c_v \sum_{\beta \in N(v)} (\mu_{x,y,\beta \rightarrow v}(\hat{y}_v) + \lambda_{x,y,v \rightarrow \beta}(\hat{y}_v)) = \left(\sum_{\beta \in N(v)} c_\beta \right) \left(\phi_{x,y,v}(\hat{y}_v) - \sum_{\beta \in N(v)} \lambda_{x,y,v \rightarrow \beta}(\hat{y}_v) \right). \quad (11)$$

We wish to find the optimal value of $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$, namely the value that satisfies Eq. (10). For that purpose we recover the value of $\sum_{\beta \in N(v)} \lambda_{x,y,v \rightarrow \beta}(\hat{y}_v)$ from (11):

$$\tilde{c}_{x,y,v \rightarrow \alpha} + \left(c_v + \sum_{\beta \in N(v)} c_\beta \right) \left(\sum_{\beta \in N(v)} \lambda_{x,y,v \rightarrow \beta}(\hat{y}_v) \right) = \left(\sum_{\beta \in N(v)} c_\beta \right) \phi_{x,y,v}(\hat{y}_v) - c_v \sum_{\beta \in N(v)} \mu_{x,y,\beta \rightarrow v}(\hat{y}_v).$$

Plugging this into 10 gives

$$\mu_{x,y,\alpha \rightarrow v}(\hat{y}_v) + \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v) = \frac{c_\alpha}{c_v + \sum_{\beta \in N(v)} c_\beta} \left(\phi_{x,y,v}(\hat{y}_v) + \sum_{\beta \in N(v)} \mu_{x,y,\beta \rightarrow v}(\hat{y}_v) \right) + c_{x,y,v \rightarrow \alpha}$$

which concludes the proof for $\epsilon, c_\alpha, c_v > 0$. Whenever any of these quantities is zero, Danskin's theorem (cf. Bertsekas et al. (2003), Theorem 4.5.1) states that its corresponding subgradient is described by a probability distribution over its maximal assignments. Therefore if $c_\alpha = 0$ in the objective function, then equality (10) holds for every c_α , and similarly whenever $c_v = 0$ in the objective, equality holds for every c_v . \square

It is computationally appealing to find the optimal $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$. When the optimal value cannot be found, one usually takes a step in the direction of the negative gradient and the objective function needs to be computed to ensure that the chosen step size reduces the objective. Obviously, computing the objective function at every iteration significantly slows the algorithm. Since the optimal $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ can be found, the block coordinate descent algorithm can be executed efficiently in distributed manner, as every $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ is computed independently. The only interactions occur when computing the normalization step $c_{x,y,v \rightarrow \alpha}$. This allows for easy computation in GPUs.

We now turn to describe how to change θ in order to improve the approximated structured prediction. Since we cannot find the optimal θ_r while holding the rest fixed, we perform a step in the direction of the negative gradient. We choose the step size η to guarantee a decent on the objective.

Lemma 4 *The gradient of the approximated structured prediction program in Theorem 2 with respect to θ_r equals to*

$$\sum_{(x,y) \in \mathcal{S}, v \in V_{r,x}, \hat{y}_v} b_{x,y,v}(\hat{y}_v) \phi_{r,v}(x, \hat{y}_v) + \sum_{(x,y) \in \mathcal{S}, \alpha \in E_{r,x}, \hat{y}_\alpha} b_{x,y,\alpha}(\hat{y}_\alpha) \phi_{r,\alpha}(x, \hat{y}_\alpha) - d_r + C \cdot |\theta_r|^{p-1} \cdot \text{sign}(\theta_r),$$

where

$$b_{x,y,v}(\hat{y}_v) \propto \exp \left(\frac{e_y(\hat{y}_v) + \sum_{r:v \in V_{r,x}} \theta_r \phi_{r,v}(x, \hat{y}_v) - \sum_{\alpha \in N(v)} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{\epsilon c_v} \right)$$

$$b_{x,y,\alpha}(\hat{y}_\alpha) \propto \exp \left(\frac{\sum_{r:\alpha \in E_{r,x}} \theta_r \phi_{r,\alpha}(x, \hat{y}_\alpha) + \sum_{v \in N(\alpha)} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{\epsilon c_\alpha} \right)$$

However, if either ϵ and/or c_α equal zero, then the beliefs $b_{x,y,\alpha}(\hat{y}_\alpha)$ can be taken from the set of probability distributions over support of the max-beliefs, namely $b_{x,y,\alpha}(\hat{y}_\alpha^*) > 0$ only if $\hat{y}_\alpha^* \in \text{argmax}_{\hat{y}_\alpha} \left\{ \sum_{r:\alpha \in E_{r,x}} \theta_r \phi_{r,\alpha}(x, \hat{y}_\alpha) + \sum_{v \in N(\alpha)} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v) \right\}$. Similarly for $b_{x,y,v}(\hat{y}_v^*)$ whenever ϵ and/or c_v equal zero.

Proof: This is a direct computation of the gradient. In the special case of $\epsilon, c_\alpha = 0$ then $b_{x,y,\alpha}(\hat{y}_\alpha)$ corresponds to the subgradient and similarly when $\epsilon, c_v = 0$, (Danskin's theorem, Bertsekas et al. (2003), Theorem 4.5.1). \square

The computational complexity of the gradient depends on the structure of the features. Since the value of the gradient depends on the beliefs for every $v \in V_{r,x}$ and $\alpha \in E_{r,x}$, its computation takes $|V_{r,x}| + |E_{r,x}|$ operations. Although this is a major improvement over existing methods, it is clear that our framework prefers many features with small graphical models rather than few features with large graphical models. Another computational issue relates the step size. In general, the coordinate descent scheme verifies that the chosen step size η reduces the objective. Theoretically, for $\epsilon, c_\alpha, c_i > 0$ and $p = 2$ we can use the fact that the gradient is Lipschitz to predetermine a step size η that guarantees descent. However, in practice it gives worse performance than searching for the step size.

Lemmas 3 and 4 describe the coordinate descent algorithm for the approximated structured prediction in Theorem 2. Figure 1 depicts a summary of the algorithm in the belief propagation format, setting $n_{x,y,v \rightarrow \alpha}(\hat{y}_v) = \exp \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ and $m_{x,y,\alpha \rightarrow v}(\hat{y}_v) = \exp \mu_{x,y,\alpha \rightarrow v}(\hat{y}_v)$.

The coordinate descent algorithm is guaranteed to converge, as it monotonically decreases the approximated structured prediction objective in Theorem 2, which is lower bounded by its dual program. However, convergence to the global minimum cannot be guaranteed in all cases. In particular, for $\epsilon = 0$ the coordinate descent on the approximated structured SVMs is not guaranteed to converge to its global minimum, unless one

Message-Passing algorithm for Approximated Structured Prediction:

Set $\bar{e}_{y,v}(\hat{y}_v) = \exp(e_{y,v}(\hat{y}_v))$ and similarly $\bar{\phi}_{r,v}, \bar{\phi}_{r,\alpha}$.

1. For $t = 1, 2, \dots$

(a) For every $v = 1, \dots, n$, every $(x, y) \in \mathcal{S}$, every $\alpha \in N(v)$, every $\hat{y}_v \in \mathcal{Y}_v$ do:

$$m_{x,y,\alpha \rightarrow v}(\hat{y}_v) = \left\| \prod_{r:\alpha \in E_r} \bar{\phi}_{r,\alpha}^{\theta_r}(x, \hat{y}_\alpha) \prod_{u \in N(\alpha) \setminus v} n_{x,y,u \rightarrow \alpha}(\hat{y}_u) \right\|_{1/\epsilon c_\alpha}$$

$$n_{x,y,v \rightarrow \alpha}(\hat{y}_v) \propto \left(\bar{e}_{y,v}(\hat{y}_v) \prod_{r:v \in V_r} \bar{\phi}_{r,v}^{\theta_r}(x, \hat{y}_r) \prod_{\beta \in N(v)} m_{x,y,\beta \rightarrow v}(\hat{y}_\beta) \right)^{c_\alpha/\bar{c}_v} / m_{x,y,\alpha \rightarrow v}(\hat{y}_v)$$

(b) For every $r = 1, \dots, d$ do:

For every $(x, y) \in \mathcal{S}$, every $v \in V_{r,x}$, $\alpha \in E_{r,x}$, every $\hat{y}_v \in \mathcal{Y}_v$, $\hat{y}_\alpha \in \mathcal{Y}_\alpha$ set:

$$b_{x,y,v}(\hat{y}_v) \propto \left(\bar{e}_{y,v}(\hat{y}_v) \prod_{r:v \in V_{r,x}} \bar{\phi}_{r,v}^{\theta_r}(x, \hat{y}_r) \prod_{\alpha \in N(v)} n_{x,y,v \rightarrow \alpha}^{-1}(\hat{y}_v) \right)^{1/\epsilon c_v}$$

$$b_{x,y,\alpha}(\hat{y}_\alpha) \propto \left(\prod_{r:\alpha \in E_{r,x}} \bar{\phi}_{r,\alpha}^{\theta_r}(x, \hat{y}_\alpha) \prod_{v \in N(\alpha)} n_{x,y,v \rightarrow \alpha}(\hat{y}_v) \right)^{1/\epsilon c_\alpha}$$

$$\theta_r \leftarrow \theta_r - \eta \left(\sum_{(x,y) \in \mathcal{S}, v \in V_{r,x}, \hat{y}_v} b_{x,y,v}(\hat{y}_v) \phi_{r,v}(x, \hat{y}_v) + \sum_{(x,y) \in \mathcal{S}, \alpha \in E_{r,x}, \hat{y}_\alpha} b_{x,y,\alpha}(\hat{y}_\alpha) \phi_{r,\alpha}(x, \hat{y}_\alpha) - c_r + C \cdot |\theta_r|^{p-1} \cdot \text{sign}(\theta_r) \right)$$

Figure 1: The block coordinate descent algorithm for approximated structured prediction in Theorem 2, as described in lemmas 3, 4.

use subgradient methods which are not monotonically decreasing. Moreover, even when we are guaranteed to converge to the global minimum, when $\epsilon, c_\alpha, c_v > 0$, the sequence of variables $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ generated by the algorithm is not guaranteed to converge to an optimal solution, nor to be bounded. As a trivial example, adding an arbitrary constant to the variables, $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v) + c$, does not change the objective value, hence the algorithm can generate monotonically decreasing unbounded sequences. However, the beliefs generated by the algorithm are bounded and guaranteed to converge to the unique solution of the dual approximated structured prediction problem. We now summarize the convergence properties.

Claim 2 *The block coordinate descent algorithm in lemmas 3 and 4 monotonically reduces the approximated structured prediction objective in Theorem 2, therefore the value of its objective is guaranteed to converge. Moreover, if $\epsilon, c_\alpha, c_v > 0$, the objective is guaranteed to converge to the global minimum, and its sequence of beliefs are guaranteed to converge to the unique solution of the approximated structured prediction dual.*

Proof: The approximated structured prediction dual is strictly concave in the dual variables $b_{x,y,v}(\hat{y}_v), b_{x,y,\alpha}(\hat{y}_\alpha), \mathbf{z}$ subject to linear constraints. The claim properties are a direct consequence of Tseng and Bertsekas (1987) for this type of programs. \square

The convergence result has a practical implication, describing the ways we can estimate the convergence of the algorithm, either by the primal objective, the dual objective or the beliefs. The approximated structured prediction can also be used for non-concave entropy approximations, such as the Bethe entropy, where $c_\alpha > 0$ and $c_v < 0$. In this case the algorithm is well defined, and its stationary points correspond to the stationary points of the approximated structured prediction and its dual. Intuitively, this statement holds since the coordinate descent algorithm iterates over points $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v), \theta_r$ with vanishing gradients. Equivalently the algorithm iterates over saddle points $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v), b_{x,y,v}(\hat{y}_v), b_{x,y,\alpha}(\hat{y}_\alpha)$ and θ_r, z_r of the Lagrangian defined in Theorem 2. Whenever the dual program is concave these saddle points are optimal points of the convex primal, but for non-concave dual the algorithm iterates over saddle points. This is summarized in the claim below:

Claim 3 *Whenever the approximated structured prediction is not convex, i.e., $\epsilon, c_\alpha > 0$ and $c_v < 0$, the algorithm in lemmas 3 and 4 is not guaranteed to converge, but whenever it converges it reaches a stationary point of the primal and dual approximated structured prediction programs.*

Proof: The approximated structured prediction in Theorem 2 is unconstrained. The update rules defined in Lemmas 3 and 4 are directly related to vanishing points of the gradient of this function, even when it is non-convex. Therefore a stationary point of the algorithm corresponds to an assignment $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v), \theta_r$ for which the gradient equals zero, or equivalently a stationary point of the approximated structured prediction.

The dual approximated structured prediction in (9) is a constrained optimization and its stationary points are saddle points of the Lagrangian, defined in Theorem 2, with respect to the probability simplex $b_{x,y,v}(\hat{y}_v) \in \Delta_{\mathcal{Y}_v}$ and $b_{x,y,\alpha}(\hat{y}_\alpha) \in \Delta_{\mathcal{Y}_\alpha}$. Note that since $\epsilon, c_\alpha, c_v \neq 0$ the entropy functions act as barrier functions on the nonnegative cone, therefore we need not consider the nonnegative constraints over the beliefs. In the following we show that at stationary points the inferred beliefs of the Lagrangian satisfy the marginalization constraints, therefore are saddle points of the Lagrangian.

When $\epsilon, c_\alpha > 0$ the stationary beliefs $b_{x,y,\alpha}(\hat{y}_\alpha)$ are achieved by maximizing over $\Delta_{\mathcal{Y}_\alpha}$, resulting in

$$b_{x,y,\alpha}(\hat{y}_\alpha) \propto \exp \left(\frac{\sum_{r:\alpha \in E_{r,x}} \theta_r \phi_{r,\alpha}(x, \hat{y}_\alpha) + \sum_{v \in N(\alpha)} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{\epsilon c_\alpha} \right).$$

However, since $c_v < 0$ the stationary beliefs $b_{x,y,v}(\hat{y}_v)$ are achieved by *minimizing* over $\Delta_{\mathcal{Y}_v}$ resulting in

$$b_{x,y,v}(\hat{y}_v) \propto \exp \left(\frac{e_y(\hat{y}_v) + \sum_{r:v \in V_{r,x}} \theta_r \phi_{r,v}(x, \hat{y}_v) - \sum_{\alpha \in N(v)} \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{\epsilon c_v} \right).$$

To prove these beliefs correspond to a stationary point we show that they satisfy the marginalization constraints. This fact is a direct consequence of the update rule in Lemma

3, where by direct computation one can verify that

$$\sum_{\hat{y}_\alpha \setminus \hat{y}_v} b_{x,y,\alpha}(\hat{y}_\alpha) \propto \exp \left(\frac{\mu_{x,y,\alpha \rightarrow v}(\hat{y}_v) + \lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)}{\epsilon c_\alpha} \right).$$

Following the definition of $b_{x,y,v}(\hat{y}_v)$ one can see that the update rule in Lemma 3 enforces the marginalization constraints. This implies that the gradient of the approximated structured prediction program measures the disagreements between $\sum_{\hat{y}_\alpha \setminus \hat{y}_v} b_{x,y,\alpha}(\hat{y}_\alpha)$ and $b_{x,y,v}(\hat{y}_v)$, and the gradient vanishes only when they agree. Therefore these beliefs correspond to a saddle point of the Lagrangian. \square

The order of the updates in the algorithm in Figure 1 is not important to guarantee the convergence properties in Claims 2, 3. For example, one can perform the updates of the messages $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ until no changes can be made, resulting in beliefs which agree on their marginal probabilities, and then perform an update step for θ_r . This method is closely related to the heuristic for solving structured prediction tasks, namely CRFs and structured SVMs, with approximated inference engine. This heuristic runs an approximate inference engine to infer beliefs which agree on their marginal probabilities, and use them to update the θ_r . However, there are two important differences between these two approach in their accuracy and efficiency: The algorithm in Figure 1 solves the approximated structured prediction accurately, since it finds a step size η for the update of θ_r that reduces the approximated structured prediction objective (8). On the other hand, when using the approximate inference heuristic, one cannot determine a step size η to reduce the CRFs and structured SVMs objectives, since these objectives cannot be computed accurately for graph with cycles. The algorithm in Figure 1 is also more efficient from the structured prediction heuristic, since it describes a way to update θ_r even if the inferred beliefs do not agree on their marginal probabilities, or equivalently $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ did not reach a stationary point. This is based on our theoretical framework in Lemmas 3, 4, which supports performing small number of approximate inference updates of $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$. These updates re-uses the values of previous iterations to extract intermediate beliefs $b_{x,y,v}(\hat{y}_v)$, $b_{x,y,\alpha}(\hat{y}_\alpha)$, which not necessarily agree on their marginal probabilities, in order to optimize θ_r . This is in contrast to running the approximated inference heuristic, which do not have a principled way to re-use previous computations and its beliefs are used for optimizing θ_r only after convergence, which is computationally intensive as a subroutine.

5. Experimental evaluation

We performed experiments on 2D grids since they are widely used to represent images, and have many cycles. We first investigate the role of ϵ in the accuracy and running time of our algorithm, for fixed $c_\alpha, c_v = 1$. We used a 10×10 binary image and randomly generated 10 corrupted samples flipping every bit with 0.2 probability. We trained the model using $\epsilon = \{1, 0.5, 0.01, 0\}$, ranging from approximated CRFs ($\epsilon = 1$) to approximated structured SVM ($\epsilon = 0$) and its smooth version ($\epsilon = 0.01$). The runtimes are 323, 324, 326, 294 seconds for $\epsilon = \{1, 0.5, 0.01, 0\}$ respectively. As ϵ gets smaller the runtime slightly increases, but it decreases for $\epsilon = 0$ since the ℓ_∞ norm is efficiently computed using the max function. However, $\epsilon = 0$ is less accurate than $\epsilon = 0.01$; When the approximated structured SVM

	Gaussian noise				Bimodal noise			
	I_1	I_2	I_3	I_4	I_1	I_2	I_3	I_4
LBP-SGD	2.7344	2.4707	3.2275	2.3193	5.2905	4.4751	6.8164	7.2510
LBP-SMD	2.7344	2.4731	3.2324	2.3145	5.2954	4.4678	6.7578	7.2583
LBP-BFGS	2.7417	2.4194	3.1299	2.4023	5.2148	4.3994	6.0278	6.6211
MF-SGD	3.0469	3.0762	4.1382	2.9053	10.0488	41.0718	29.6338	53.6035
MF-SMD	2.9688	3.0640	3.8721	14.4360	–	–	–	–
MF-BFGS	3.0005	2.7783	3.6157	2.4780	5.2661	4.6167	6.4624	7.2510
Ours	0.0488	0.0073	0.1294	0.1318	0.0537	0.0244	0.1221	0.9277

Figure 2: **Gaussian and bimodal noise:** Comparison of our approach to loopy belief propagation and mean field approximations when optimizing using BFGS, SGD and SMD. Note that our approach significantly outperforms all the baselines. MF-SMD did not work for Bimodal noise.

converges, the gap between the primal and dual objectives was 1.3, and only 10^{-5} for $\epsilon > 0$. This is to be expected since the approximated structured SVM is non-smooth (Claim 2).

We generated test images in a similar fashion. When using the same ϵ for training and testing we obtained 2 misclassifications for $\epsilon > 0$ and 109 for $\epsilon = 0$. We conjecture that this comes from the non-zero primal-dual gap of $\epsilon = 0$. We also evaluated the quality of the solution using different values of ϵ for training and inference, following Wainwright (2006). When predicting with smaller ϵ than the one used for learning the results are marginally worse than when predicting with the same ϵ . However, when predicting with larger ϵ , the results get significantly worse, e.g., learning with $\epsilon = 0.01$ and predicting with $\epsilon = 1$ results in 10 errors, and only 2 when $\epsilon = 0.01$.

The main advantage of our algorithm is that it can efficiently learn many parameters in a graphical model. We now compared, in a similarly generated dataset of size 5×5 , a model learned with different parameters for every edge and vertex (≈ 300 parameters) and a model learned with parameters shared among the vertices and edges (2 parameters for edges and 2 for vertices) used by Kumar and Hebert (2003). Using large number of parameters increases performance: sharing parameters resulted in 16 misclassifications, while optimizing over the 300 parameters resulted in 2 errors. Our algorithm avoids overfitting in this case, we conjecture it is due to the regularization.

We now compare our approach to state-of-the-art CRF solvers. We use the binary image dataset of Kumar and Hebert (2003) that consists of 4 different 64×64 base images. Each base image was corrupted 50 times with each type of noise. Following Vishwanathan et al. (2006), we trained different models to denoise each individual image, using 40 examples for training and 10 for test. We compare our approach to the result of approximating the conditional likelihood using loopy belief propagation (LBP) and mean field approximation (MF). For each of these approximations, we use stochastic gradient descent (SGD), stochastic meta-descent (SMD) and BFGS to learn the parameters. We do not report pseudolikelihood (PL) results since it did not work. Note that the same behavior of PL was noticed by Vishwanathan et al. (2006). To reduce the computational complexity and the chances of convergence, Kumar and Hebert (2003); Vishwanathan et al. (2006) forced their parameters to be shared across all nodes such that $\forall i, \theta_i = \theta^{(n)}$ and $\forall i, \forall j \in N(i), \theta_{ij} = \theta^e$. In contrast,

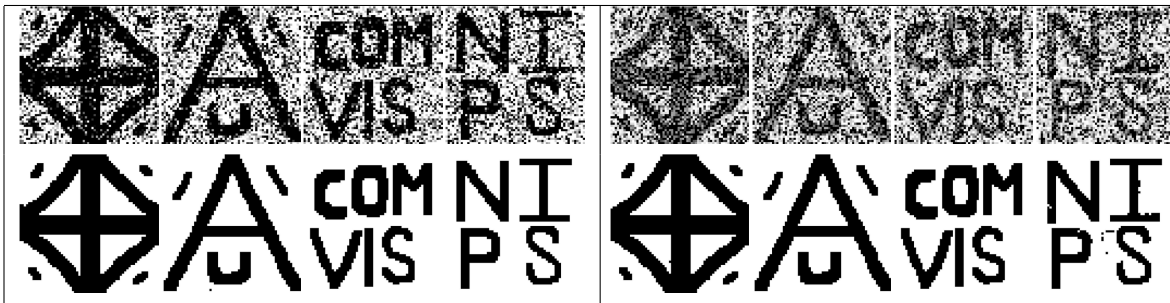


Figure 3: **Denoising results:** Gaussian (left) and Bimodal (right) noise.

since our approach is efficient, we can exploit the full flexibility of the graph and learn more than 10,000 parameters. Note that this is computationally prohibitive with the baselines. For the local features we simply use the pixel values, and for the node potentials we use an Ising model with only bias features such that $\phi_{i,j} = [1, -1; -1, 1]$. For all experiments we use $\epsilon = 1$, and $p = 2$. For the baselines, we use the code, features and optimal parameters of Vishwanathan et al. (2006).

Under the first noise model, each pixel was corrupted via i.i.d. Gaussian noise with mean 0 and standard deviation of 0.3. Fig. 2 depicts test error in (%) for the different base images (i.e., I_1, \dots, I_4). Note that our approach outperforms considerably the loopy belief propagation and mean field approximations for all optimization criteria (BFGS, SGD, SMD). For example, for the first base image the error of our approach is 0.0488%, which is equivalent to a 2 pixels error on average. In contrast the best baseline gets 112 pixels wrong on average. Fig. 3 (left) depicts test examples as well as our denoising results. Note that our approach is able to cope with large amounts of noise.

Under the second noise model, each pixel was corrupted with an independent mixture of Gaussians. For each class, a mixture of 2 Gaussians with equal mixing weights was used, yielding the Bimodal noise. The mixture model parameters were (0.08, 0.03) and (0.46, 0.03) for the first class and (0.55, 0.02) and (0.42, 0.10) for the second class, with (a, b) a Gaussian with mean a and standard deviation b . Fig. 2 depicts test error in (%) for the different base images. As before, our approach outperforms all the baselines. We do not report MF-SMD results since it did not work. Denoised images are shown in Fig. 3 (right). We now show how our algorithm converges in a few iterations. Fig. 4 depicts the primal and dual training errors as a function of the number of iterations. Note that our algorithm converges, and the dual and primal values are very tight after a few iterations.

6. Related work

We now discuss related work. For the special case of CRFs, the idea of approximating the entropy function with local entropies was used by Wainwright (2006); Ganapathi et al. (2008). In particular, Wainwright (2006) proved that using a concave entropy approximation gives robust prediction. Ganapathi et al. (2008) used the non-concave Bethe entropy approximation $c_\alpha = 1, c_v = 1 - |N(v)|$ as well as the concave approximation $c_\alpha = 1, c_v = 0$. Our work differs from these works in two aspects: we derive an efficient algorithm in Sec-

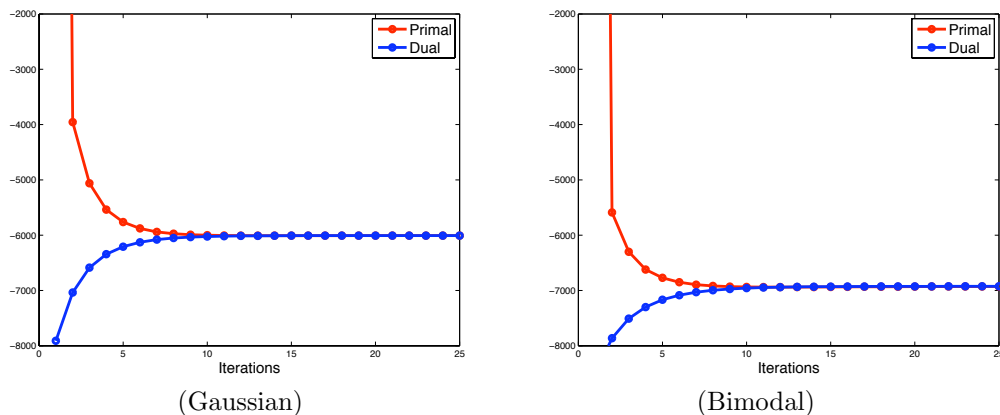


Figure 4: **Convergence.** Primal and dual train errors when for I_1 is corrupted with Gaussian and Bimodal noise. Our algorithm is able to converge in a few iterations.

tion 4 for the concave approximated program ($c_\alpha, c_v > 0$) and our framework and algorithm include structured SVMs, as well as their smooth approximation when $\epsilon \rightarrow 0$.

Some forms of approximated structured prediction were investigated for the special case of CRFs. Sutton and McCallum (2009) described a similar program, but without the Lagrange multipliers $\lambda_{x,y,v \rightarrow \alpha}(\hat{y}_v)$ and no regularization, i.e., $C = 0$. As a result the local log-partition functions are independent, and efficient counting algorithm can be used for learning. Ganapathi et al. (2008) derived an approximated program for $c_\alpha = 1, c_v = 0$ without regularization which was solved by the BFGS convex solver. Also, the constraints of Ganapathi et al. (2008) were composed differently which lead to a different dual formulation. Our work is different as it considers efficient algorithms for approximated structured prediction, and takes advantage of the graphical model by sending messages along its edges. We show in the experiments that this significantly improves the run-time of the algorithm. Also, our approximated structured prediction includes as special cases approximated CRF, for $\epsilon = 1$, and approximated structured SVM, for $\epsilon = 0$. Moreover, we describe how to smoothly approximate the structured SVMs to avoid the shortcomings of subgradient methods, by simply setting $\epsilon \rightarrow 0$.

7. Conclusion and Discussion

In this paper we have related CRFs and structured SVMs and shown that the soft-max, a variant of the log-partition function, approximates smoothly the structured SVM hinge loss. We have also proposed an approximation for structured prediction problems based on local entropy approximations and derived an efficient message-passing algorithm that is guaranteed to converge, even for general graphs. We have demonstrated the effectiveness of our approach to learn graphs with large number of parameters in an image denoising task. In the future we plan to investigate other domains of application such as image segmentation.

References

- D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.
- M. Collins, A. Globerson, T. Koo, X. Carreras, and P.L. Bartlett. Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. *The Journal of Machine Learning Research*, 9:1775–1822, 2008.
- W. Fenchel. *Convex cones, sets, and functions*. Princeton University, Department of Mathematics, 1951.
- T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. In *Proceedings of the 25th international conference on Machine learning*, pages 304–311. ACM, 2008.
- V. Ganapathi, D. Vickrey, J. Duchi, and D. Koller. Constrained approximate maximum entropy learning of markov random fields. In *Uncertainty in Artificial Intelligence*, 2008.
- T. Hazan and A. Shashua. Norm-Product Belief Propagation: Primal-Dual Message-Passing for Approximate Inference. *Arxiv preprint arXiv:0903.3127*, 2009.
- T. Heskes. Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies. *Journal of Artificial Intelligence Research*, 26(1):153–190, 2006.
- J.K. Johnson, D.M. Malioutov, and A.S. Willsky. Lagrangian relaxation for MAP estimation in graphical models. In *Proceedings of the Allerton Conference on Control, Communication and Computing*. Citeseer, 2007.
- S. Kumar and M. Hebert. Discriminative Fields for Modeling Spatial Dependencies in Natural Images. In *Neural Information Processing Systems*. MIT Press, Cambridge, MA, 2003.
- J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference of Machine Learning*, pages 282–289, 2001.
- G. Lebanon and J. Lafferty. Boosting and maximum likelihood for exponential models. *Advances in neural information processing systems*, 1:447–454, 2002.
- A. Levin and Y. Weiss. Learning to Combine Bottom-Up and Top-Down Segmentation. In *European Conference on Computer Vision*, 2006.
- T. Meltzer, A. Globerson, and Y. Weiss. Convergent message passing algorithms-a unifying view. In *Uncertainty in Artificial Intelligence*, 2009.
- J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.
- N. Ratliff, J.A. Bagnell, and M. Zinkevich. Subgradient methods for maximum margin structured learning. In *ICML Workshop on Learning in Structured Output Spaces*, 2006.

- R.T. Rockafellar. *Convex analysis*. Princeton university press, 1970.
- C. Sutton and A. McCallum. Piecewise training for structured prediction. *Machine Learning*, 77(2):165–194, 2009.
- B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)*, pages 895–902. Citeseer, 2002.
- B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. *Advances in neural information processing systems*, 16:51, 2004.
- B. Taskar, S. Lacoste-Julien, and M. I. Jordan. Structured prediction, dual extragradient and Bregman projections. *JMLR*, 7:1653–1684, 2006.
- P. Tseng and D.P. Bertsekas. Relaxation methods for problems with strictly convex separable costs and linear constraints. *Mathematical Programming*, 38(3):303–321, 1987.
- I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(2):1453, 2006.
- S. Vishwanathan, N. Schraudolph, M. Schmidt, and K. Murphy. Accelerated Training of Conditional Random Fields with Stochastic Meta-Descent . In *International Conference in Machine Learning*, 2006.
- P.O. Vontobel and R. Koetter. Towards low-complexity linear-programming decoding. *Arxiv preprint cs/0602088*, 2006.
- M.J. Wainwright. Estimating the Wrong Graphical Model: Benefits in the Computation-Limited Setting. *The Journal of Machine Learning Research*, 7:1859, 2006.
- M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.
- M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. MAP estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005a.
- MJ Wainwright, TS Jaakkola, and AS Willsky. A new class of upper bounds on the log partition function. *Information Theory, IEEE Transactions on*, 51(7):2313–2335, 2005b.
- W. Wiegerinck and T. Heskes. Fractional belief propagation. In *Advances in Neural Information Processing Systems 15: Proceedings of the 2002 Conference*, page 455. MIT Press, 2003.
- C. Yanover, O. Schueler-Furman, and Y. Weiss. Minimizing and learning energy functions for side-chain prediction. In *Research in Computational Molecular Biology*, pages 381–395. Springer, 2007.

JS Yedidia, WT Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory, IEEE Transactions on*, 51(7):2282–2312, 2005.